Load Library

In [ ]:
```
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading Dataset

In [ ]:
```
df= pd.read_csv("Movie.csv")
```

In [ ]:
```
#first 5 rows
df.head()
```

Out[ ]:

	Unnamed: 0	id	originalTitle	description	contentRating	budget
0	0	tt0111161	The Shawshank Redemption	A banker convicted of uxoricide forms a friend...	R	25000000.0
1	1	tt0068646	The Godfather	The aging patriarch of an organized crime dyna...	R	6000000.0
2	2	tt0468569	The Dark Knight	When a menace known as the Joker wreaks havoc ...	PG-13	185000000.0
3	3	tt0071562	The Godfather Part II	The early life and career of Vito Corleone in ...	R	13000000.0
4	4	tt0050083	12 Angry Men	The jury in a New York City murder trial is fr...	Approved	350000.0

In [ ]:
```
#last 5 rows
df.tail()
```

Out[ ]:

	Unnamed: 0	id	originalTitle	description	contentRating	budge
245	245	tt0035446	To Be or Not to Be	During the German occupation of Poland, an act...	Approved	Na
246	246	tt16492678	Kimetsu no Yaiba: Tsuzumi Yashiki-hen	Tanjiro ventures to the south-southeast where ...	NaN	Na
247	247	tt1954470	Gangs of Wasseypur	A clash between Sultan and Shahid Khan leads t...	Not Rated	184000000
248	248	tt0758758	Into the Wild	After graduating from Emory University, top st...	R	15000000
249	249	tt1454029	The Help	An aspiring author during the civil rights mov...	PG-13	25000000

Cleaning Process

In [ ]:
```
#shape of the dataset
print("Shape of Dataset: ", df.shape)
```

```
Shape of Dataset:  (250, 11)
```

In [ ]:
```
#Number of Rows & Columns
print("Number of Rows: ",df.shape[0])
print("Number of Columns: ", df.shape[1])
```

```
Number of Rows:  250
Number of Columns:  11
```

In [ ]:
```
#checking information.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250 entries, 0 to 249
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Unnamed: 0      250 non-null    int64  
 1   id              250 non-null    object 
 2   originalTitle   250 non-null    object 
 3   description     250 non-null    object 
 4   contentRating   244 non-null    object 
 5   budget          228 non-null    float64
 6   grossWorldwide  247 non-null    float64
 7   genres          250 non-null    object 
 8   isAdult         250 non-null    bool   
 9   numVotes        250 non-null    int64  
 10  averageRating   250 non-null    float64
dtypes: bool(1), float64(3), int64(2), object(5)
memory usage: 19.9+ KB
```

In [ ]:
```
#check data types.
df.dtypes
```

Out[ ]:

	0
Unnamed: 0	int64
id	object
originalTitle	object
description	object
contentRating	object
budget	float64
grossWorldwide	float64
genres	object
isAdult	bool
numVotes	int64
averageRating	float64

dtype: object

In [ ]:
```
#check null
df.isnull().sum()
```

Out[ ]:

	0
Unnamed: 0	0
id	0
originalTitle	0
description	0
contentRating	6
budget	22
grossWorldwide	3
genres	0
isAdult	0
numVotes	0
averageRating	0

dtype: int64

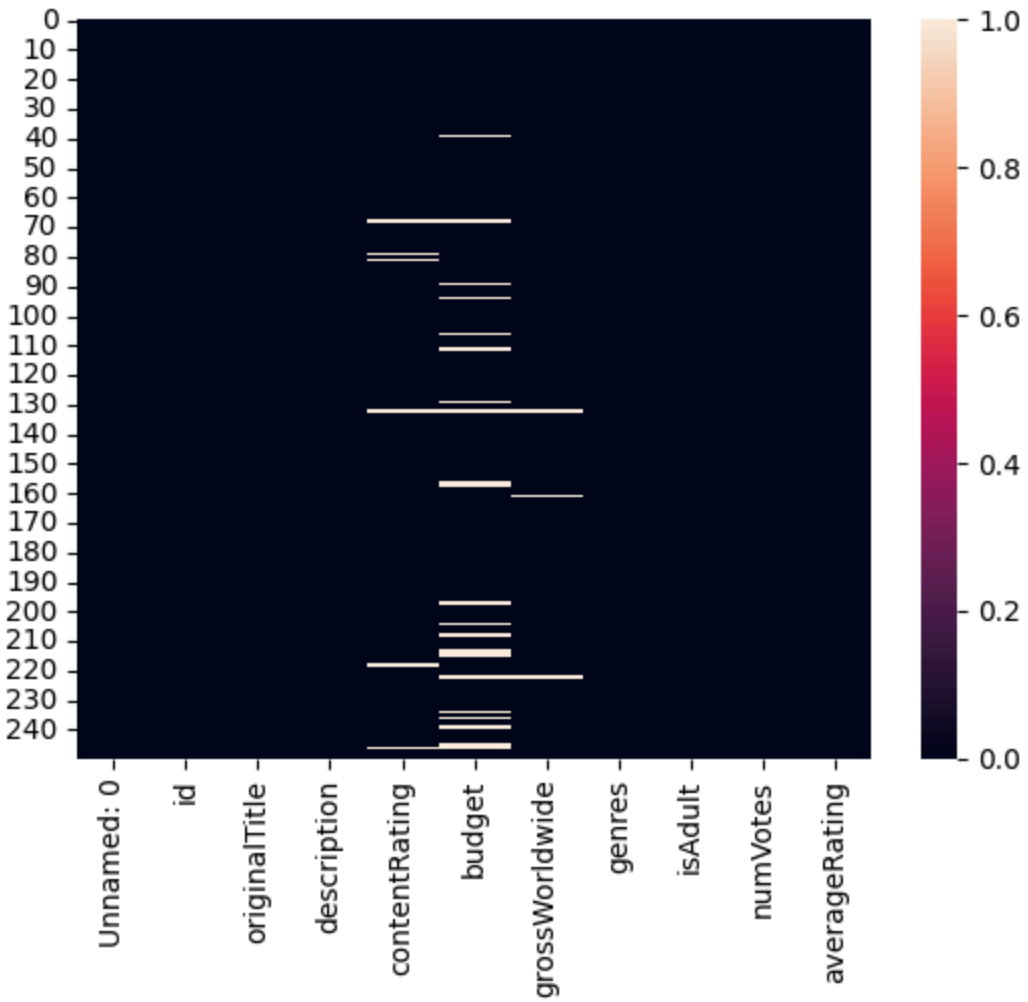
In [ ]:
```
#check null in entires data set
print(df.isnull().sum().sum())
```

31

In [ ]:
```
#check precentage wise null
print(round((df.isnull().sum().sum())/(df.shape
[0]*df.shape[1])*100,2))
```

1.13

In [ ]:
```
sns.heatmap(df.isnull())
plt.show()
```



Handling null & missing values

In [ ]:
```
df = df.dropna(subset =["contentRating","budget", "grossWorldwide"])
df.isnull().sum()
```

Out[ ]:

	0
Unnamed: 0	0
id	0
originalTitle	0
description	0
contentRating	0
budget	0
grossWorldwide	0
genres	0
isAdult	0
numVotes	0
averageRating	0

dtype: int64

Feature Engineering

In [ ]:
```
#Sucess%(ROI)
df["Sucess"] =((df["grossWorldwide"]-df["budget"])/
df["budget"])*100
df["Sucess"] = round(df["Sucess"],2)
```

In [ ]:
```
#Add verdict column
def Ver(x):
  if x<0:
    return "Flop"
  elif x >=0 and x<50:
     return "Average"
  elif x>=50 and x<100:
     return "Hit"
  elif x >=100 and x< 200:
     return "Superhit"
  elif x>=200 and x<500:
      return "Blockbuster"
  elif x>= 500:
      return "ATB" # all time blockbuster

df["Verdict"] = df["Sucess"].apply(Ver)
```

In [ ]:
```
#Add loss/profit column
def Loss_pr(a):
     if a <0:
        return "Loss"
```

```
    else:
       return "Profit"
df["Loss/Profit"] = df["Sucess"].apply(Loss_pr)
```

In [ ]:
```
#droping unusedful columns
df.drop(columns= ["Unnamed: 0","description"], inplace= True)
```

In [ ]:
```
#check duplicated
print(df.duplicated().sum())
```

```
0
```

In [ ]:
```
df.describe()
```

Out[ ]:

	budget	grossWorldwide	numVotes	averageRating	Suces
count	2.240000e+02	2.240000e+02	2.240000e+02	224.000000	224.0000
mean	1.684934e+08	2.509088e+08	8.129960e+05	8.312054	810.2867
std	1.209748e+09	3.820699e+08	5.984732e+05	0.236543	1462.8385
min	1.330000e+05	1.388000e+03	6.405600e+04	8.000000	-99.9700
25%	3.270000e+06	1.746101e+07	3.269718e+05	8.100000	83.6150
50%	1.620000e+07	8.096491e+07	7.245990e+05	8.200000	362.8050
75%	6.000000e+07	3.572349e+08	1.145412e+06	8.400000	893.1100
max	1.500000e+10	2.799439e+09	3.100661e+06	9.300000	12113.8900

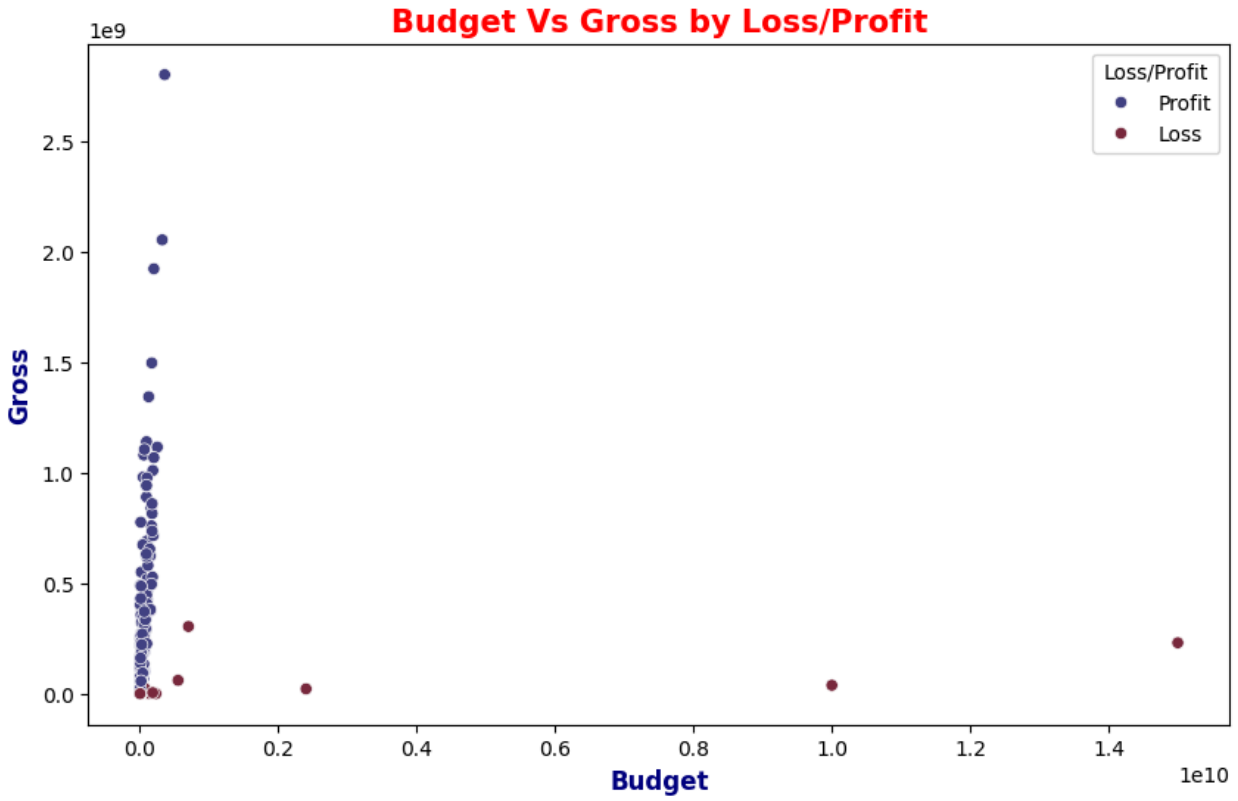
In [ ]:
```
df.describe(include= "object")
```

Out[ ]:

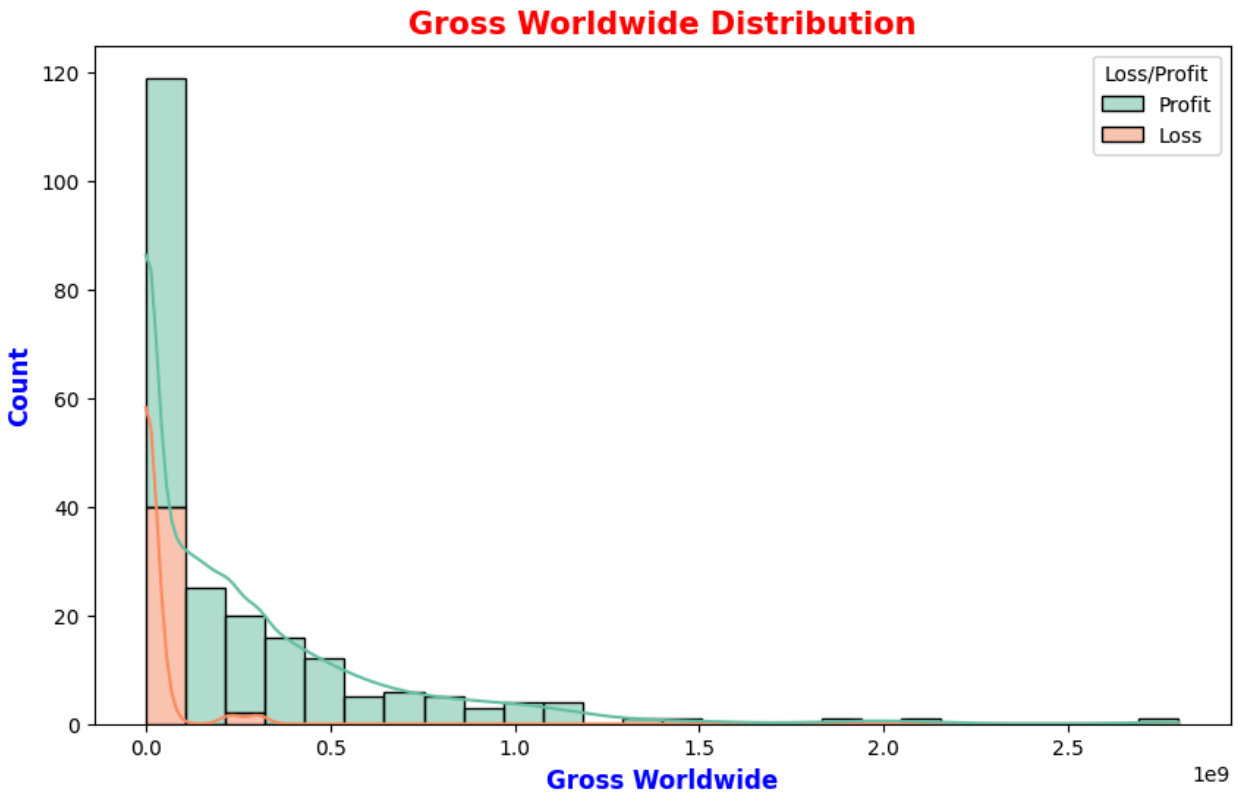
	id	originalTitle	contentRating	genres	Verdict	Loss/ Profit
count	224	224	224	224	224	224
unique	224	224	9	96	6	2
top	tt0111161	The Shawshank Redemption	R	['Drama']	ATB	Profit
freq	1	1	100	14	90	182

In [ ]:
```
#relationship between budget & Gross
plt.figure(figsize= (10,6))
sns.scatterplot(x = "budget", y ="grossWorldwide",data = df,
hue = "Loss/Profit", palette= "icefire")
plt.title("Budget Vs Gross by Loss/Profit",
color = "r", size= 15, fontweight="bold")
plt.xlabel("Budget", color="navy",
size =12,fontweight="bold")
plt.ylabel("Gross", color="navy",
```

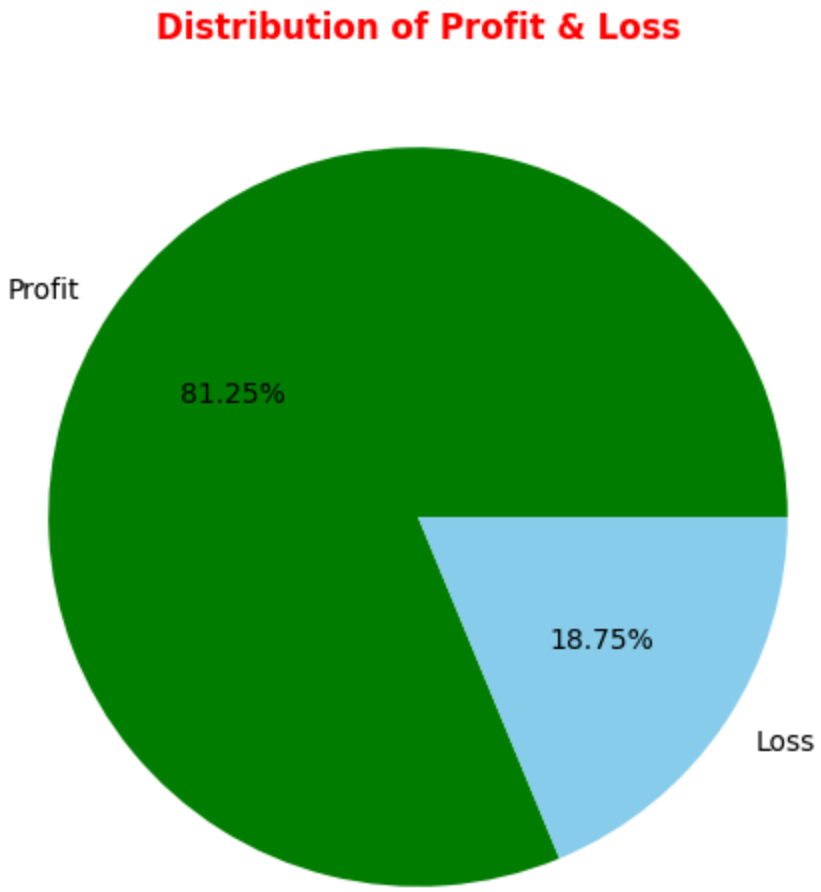
```
size =12, fontweight="bold")
plt.show()
```



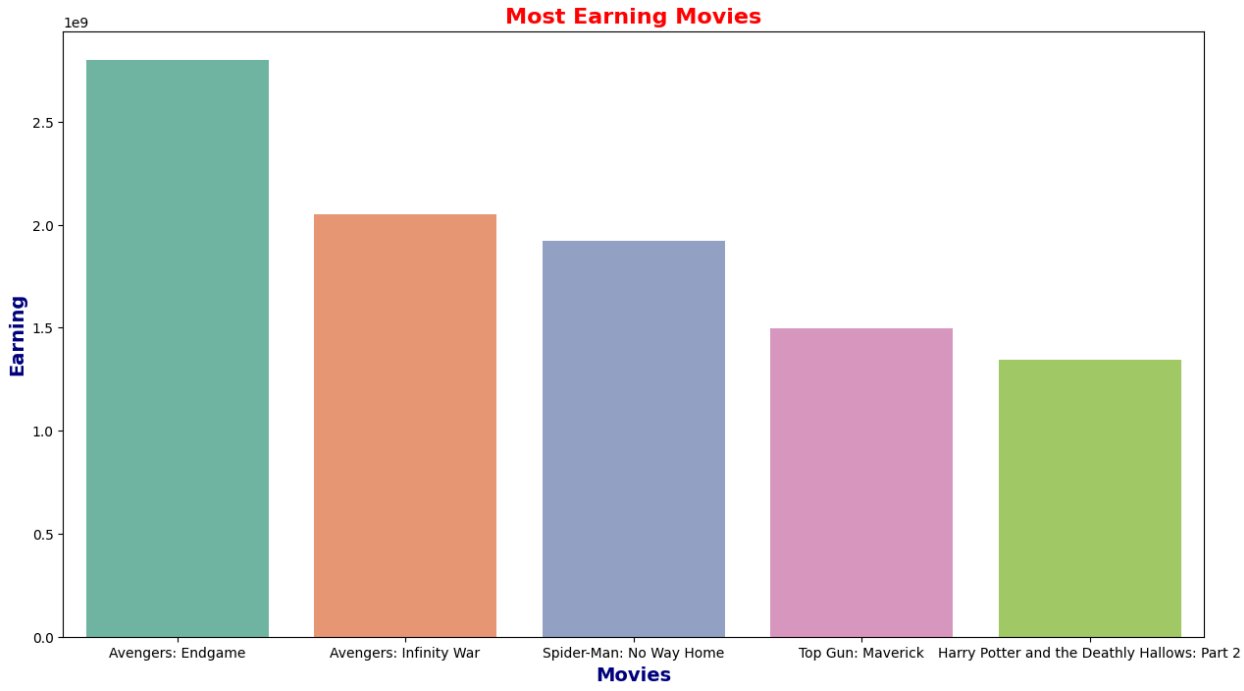
In [ ]:
```
#Gross World wide
plt.figure(figsize= (10,6))
sns.histplot(x = "grossWorldwide",data = df,
hue ="Loss/Profit",kde = True, multiple = "stack",palette="Set2")
plt.title('Gross Worldwide Distribution', color = "r",
size = 15, fontweight = "bold")
plt.xlabel("Gross Worldwide" , color= "Blue",size = 12,
 fontweight= "bold")
plt.ylabel("Count", color="Blue",
 size =12, fontweight= 'bold')
plt.show()
```



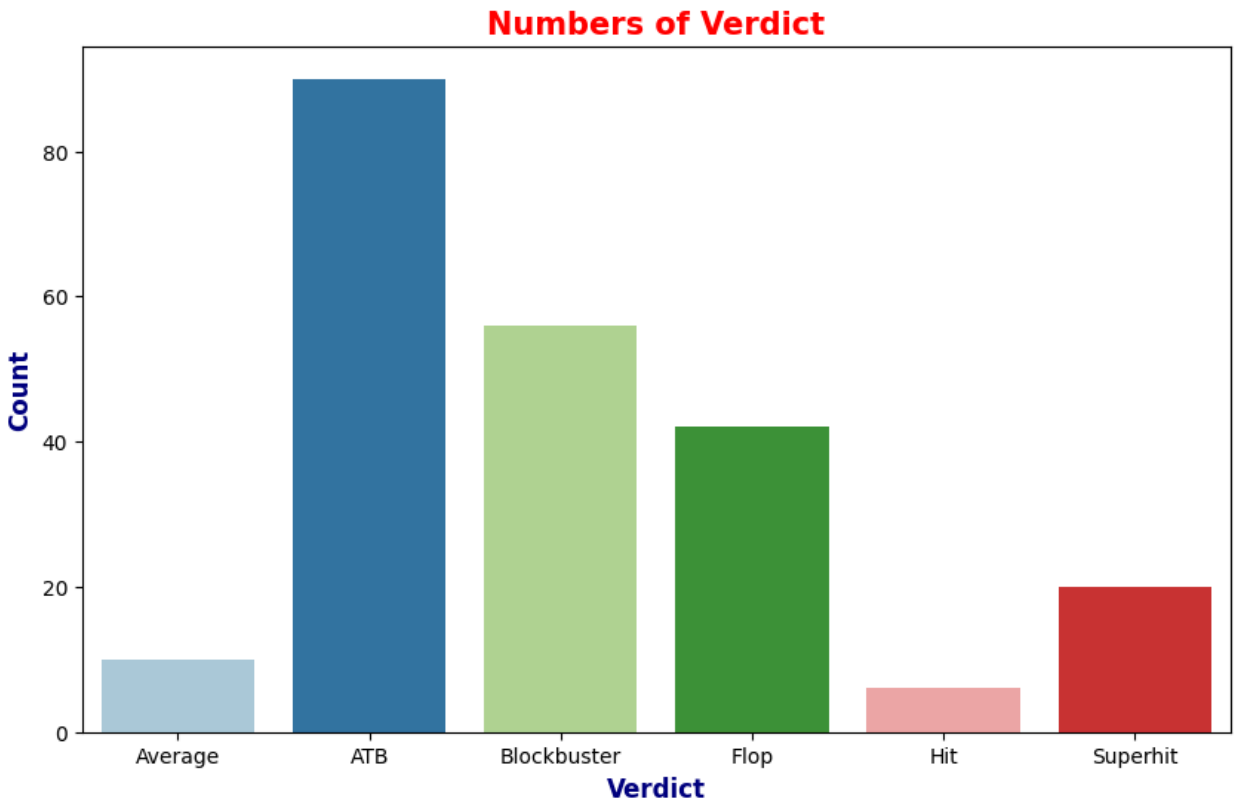
In [ ]:
```
plt.figure(figsize=(10,6))
x = df["Loss/Profit"]. value_counts()
y = df["Loss/Profit"].value_counts().keys()
plt.pie(x , labels=y,autopct = "%0.2f%%", colors= ["green", "Skyblue"])
plt.title("Distribution of Profit & Loss",
color="r",size=12, fontweight="bold")
plt.show()
```



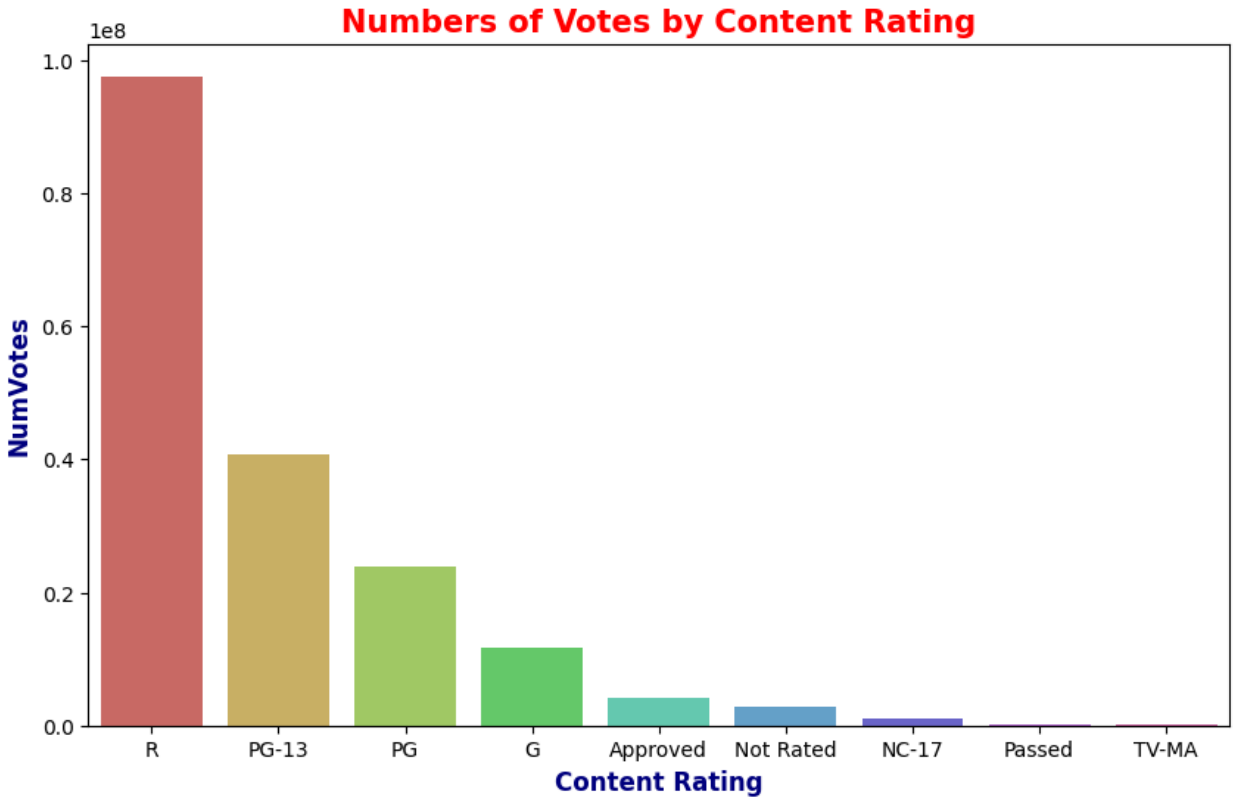
In [ ]:
```
#Top 5 highest earings Movie
plt.figure(figsize= (15,8))
df2 = df.sort_values(by = "grossWorldwide", ascending=False)[0:5]
sns.barplot(x = "originalTitle",y = "grossWorldwide",
data = df2, palette="Set2")
plt.title("Most Earning Movies", color="r",
size =16, fontweight="bold")
plt.ylabel("Earning", color="navy",
size=14, fontweight="bold")
plt.xlabel("Movies", color="navy",
size=14, fontweight="bold")
plt.show()
```



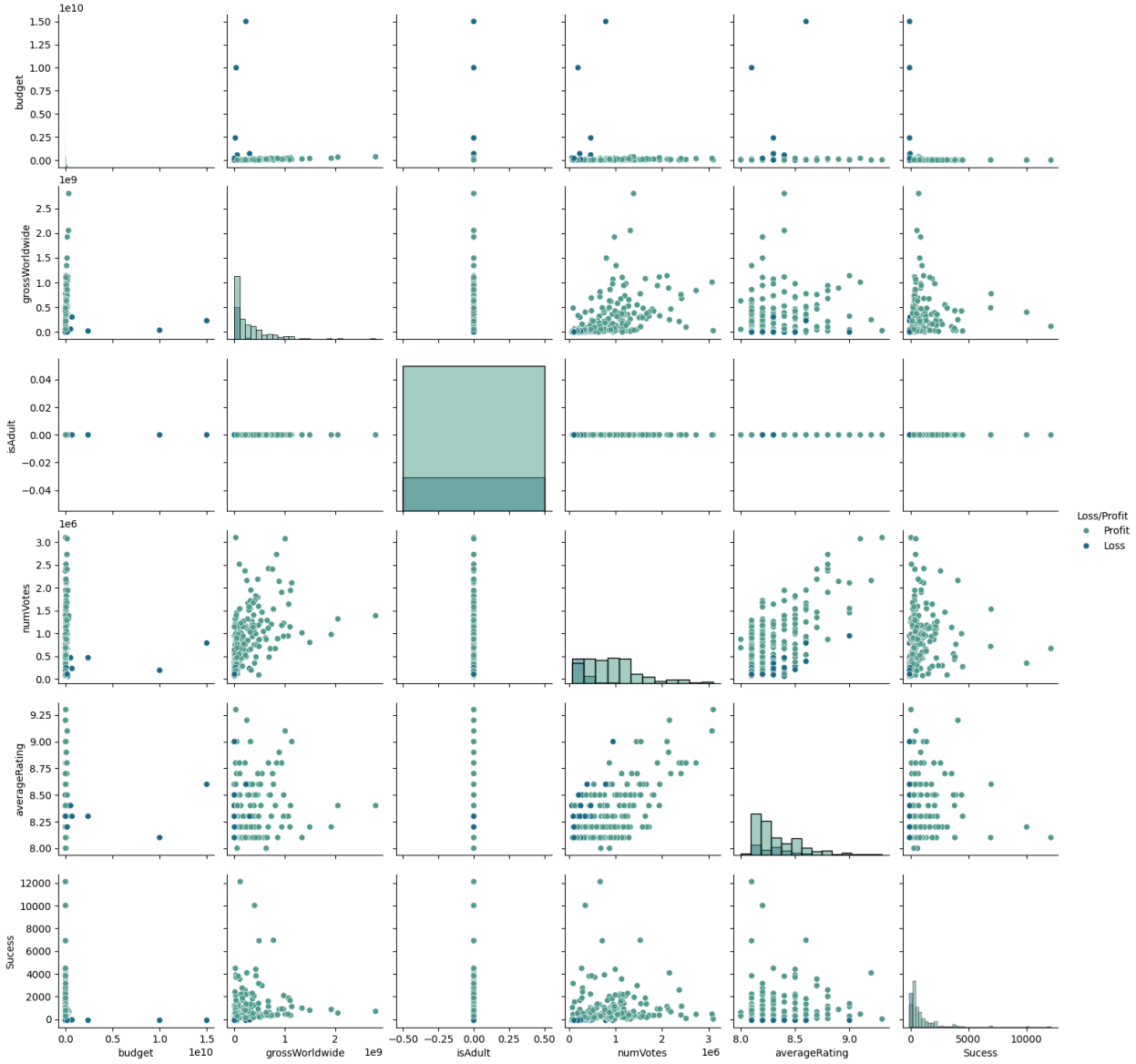
In [ ]:
```
#Count of Verdict
plt.figure(figsize= (10,6))
sns.countplot(x = "Verdict", data = df, palette="Paired")
plt.title("Numbers of Verdict",color="r",
size=15, fontweight="bold")
plt.xlabel("Verdict", color="navy",
size=12, fontweight="bold")
plt.ylabel("Count",color="navy",
size=12, fontweight="bold")
plt.show()
```



In [ ]:
```
#Sum of Numbers of Votes by content Rating
Rating= df.groupby("contentRating")["numVotes"].sum().reset_index()
Rating = Rating.sort_values(by = "numVotes", ascending=False)
plt.figure(figsize= (10,6))
sns.barplot(x = "contentRating", y = "numVotes",
data = Rating, palette="hls")
plt.title("Numbers of Votes by Content Rating",
color="r", size =15, fontweight="bold")
plt.xlabel("Content Rating", size=12,
fontweight="bold", color="navy")
plt.ylabel("NumVotes", size =12,
color="navy", fontweight="bold")
plt.show()
```



In [ ]:
```
#Understanding the relationship & Distribution
sns.pairplot(data=df, hue= "Loss/Profit",
 palette="crest" ,diag_kind = "hist")
plt.show()
```


budget
grossWorldwide
isAdult
numVotes
averageRating
Sucess
1e10
1e9
1e6
Loss/Profit
Profit
Loss